

第 3 章 构建嵌入式驱动程序开发环境

在编写驱动程序之前，需要构建一个合适的开发环境。这个环境包括合适的 Linux 操作系统、网络、交叉编译工具及 NFS 服务等。为了使读者顺利地完成开发环境的构建，本章将对这些主要内容进行讲解。

3.1 虚拟机和 Linux 安装

由于驱动开发需要涉及不同操作系统的功能，所以需要安装不同的操作系统。一般开发者习惯在 Windows 系统上安装虚拟机，然后在虚拟机上安装 Linux 系统。这种方式，可以使一台主机模拟多台主机的功能，从而提高开发的效率。这里，首先介绍安装虚拟机的方法。

3.1.1 在 Windows 上安装虚拟机

在 Window 上安装虚拟机，可以有多种选择。目前流行的虚拟机软件有 VMware 和 Virtual PC。它们都能在 Windows 系统上虚拟出多个计算机，用于安装 Linux、OS/2、FreeBSD 等其他操作系统。微软在 2003 年 2 月份收购 Connectix 后，很快发布了 Microsoft Virtual PC。但出于种种考虑，新发布的 Virtual PC 已不再明确支持 Linux、FreeBSD、NetWare、Solaris 等操作系统，只保留了 OS/2，如果要虚拟一台 Linux 计算机，只能自己手工设置。

相比而言，VMware 不论是在多操作系统的支持上，还是在执行效率上，都比 Virtual PC 明显高出一筹。所以本书选择 VMware 虚拟机构建驱动程序开发环境。从 VMware 的官方网站 <http://www.vmware.com/cn/> 可以下载到 VMware 工具，根据提示安装该软件。

建立一个虚拟机需要指定 CPU、硬盘、内存、网络、光驱等。在 VMware 中可以选择实际的物理硬盘，也可以选择用文件来模拟硬盘。在 VMware 的安装过程中，有一些特殊的地方需要注意，否则可能会造成虚拟机无法使用的现象。下面将对安装过程进行详细讲解。

(1) 启动 VMware，如图 3.1 所示。单击“起始页”的“新建虚拟机”图标，建立一台新的虚拟机。

(2) 在弹出的 New Virtual Machine Wizard 窗口中，选择“自定义（高级）”选项。这一步将对虚拟机进行自定义配置，如图 3.2 所示。单击 Next 按钮，进入下一步。

(3) 在进入的对话框中，从“硬件兼容性”下拉列表框中选择 Workstation 6.5 选项，单击 Next 按钮，如图 3.3 所示。



图 3.1 启动 VMware



图 3.2 虚拟机配置选项



图 3.3 硬件兼容性选项

(4) 进入如图 3.4 的对话框后, 选择“我将操作系统以后安装”单选按钮, 进入下一步。

(5) 在进入的对话框中选择 Linux 选项, 表示将在此虚拟机上安装 Linux 操作系统。在版本下拉列表框中选择 Other Linux 2.6.x kernel 选项, 表示将安装 2.6 内核的 Linux 系统, 如图 3.5 所示。

(6) 单击“下一步”按钮, 在进入的对话框中设置虚拟机的名字和存储位置, 如图 3.6 所示。



图 3.4 安装源选择

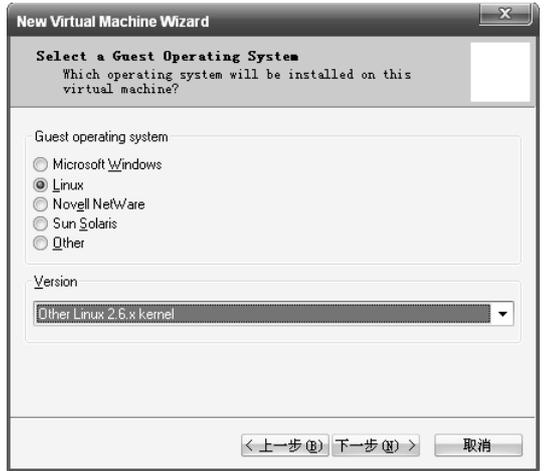


图 3.5 选择内核版本

(7) 单击 Next 按钮，进入处理器的配置阶段，选择一个处理器。因为对于嵌入式系统来说，资源往往比较有限，一般只有一个处理器，所以这里选择一个，如图 3.7 所示。



图 3.6 选择虚拟机的名字和存储位置



图 3.7 处理器个数选择

(8) 单击 Next 按钮，进入选择内存大小对话框。VMware 将会从实际的物理内存中分配指定的“虚拟机内存”大小使用。如果实际物理内存有 1GB，那么虚拟机内存选择 256MB 就可以了，如图 3.8 所示。

(9) 单击 Next 按钮，进入指定虚拟机的网络连接方式对话框。这里使用 Host-only 方式，如图 3.9 所示。具体的网络配置，将在后面讲述。

(10) 单击 Next 按钮，进入选择 I/O Adapter 对话框。这里使用默认值 (LSI Logic)，如图 3.10 所示。

(11) 单击 Next 按钮，进入下一个对话框，如图 3.11 所示。在其中选择“创建一个新的虚拟磁盘”单选按钮。

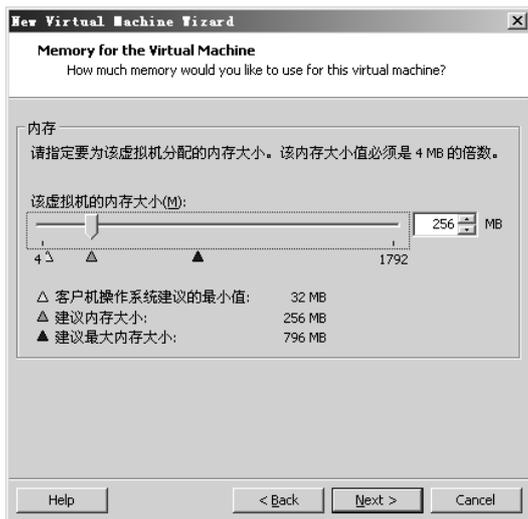


图 3.8 内存选择



图 3.9 网络连接类型



图 3.10 选择 I/O Adapter 对话框



图 3.11 选择创建新的磁盘

(12) 单击 Next 按钮，进入选择虚拟硬盘类型对话框。这里选择默认的 SCSI 硬盘，如图 3.12 所示。

(13) 单击 Next 按钮，进入指定硬盘大小对话框。这里选择 30GB，因为 Linux 开发需要的空间较大。在图 3.13 中，“以每个文件 2GB 存储虚拟磁盘”选项表示使用多个 2GB 的文件表示一个很大的虚拟硬盘。如果 Windows 的硬盘格式为 FAT32，因为其支持的最大文件是 4GB，所以要选择这个选项；如果是 NTFS 格式，则无须选择这个选项。

(14) 单击 Next 按钮，进入下一步对话框，如图 3.14 所示。在其中指定每个虚拟硬盘文件的基本名字。因为一个虚拟硬盘可能由多个文件组成，这里是每个文件都共有的名字。

(15) 单击 Next 按钮，进入下一步对话框。直接单击 Finish 按钮，就创建了一个虚拟机。

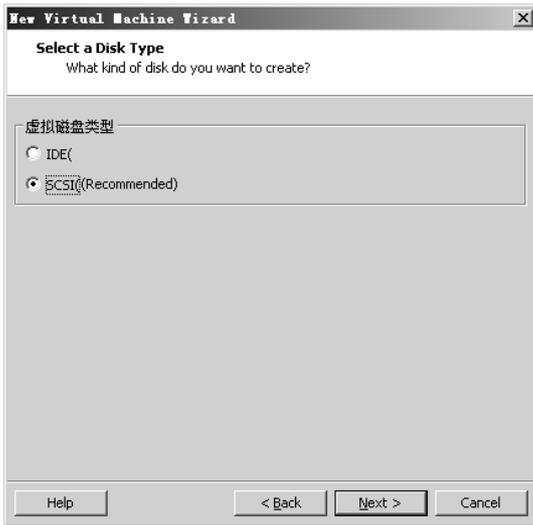


图 3.12 选择硬盘（使用默认类型）



图 3.13 指定硬盘大小



图 3.14 指定虚拟硬盘的基本名

3.1.2 在虚拟机上安装 Linux

本节介绍怎样在虚拟机上安装 Fedora 18.0，并详细介绍如何建立 Linux 开发环境。下面对安装步骤进行详细说明。

(1) 在虚拟机的光驱上选择 Fedora 18.0 的光盘镜像文件，然后启动虚拟机，进入安装界面，如图 3.15 所示。

进入安装界面后，界面中显示 Fedora 18-Beta 版，不同发行版可能介绍不一样，下面有 3 个菜单，请直接选择 Install Fedora，并回车确定，进入安装界面。

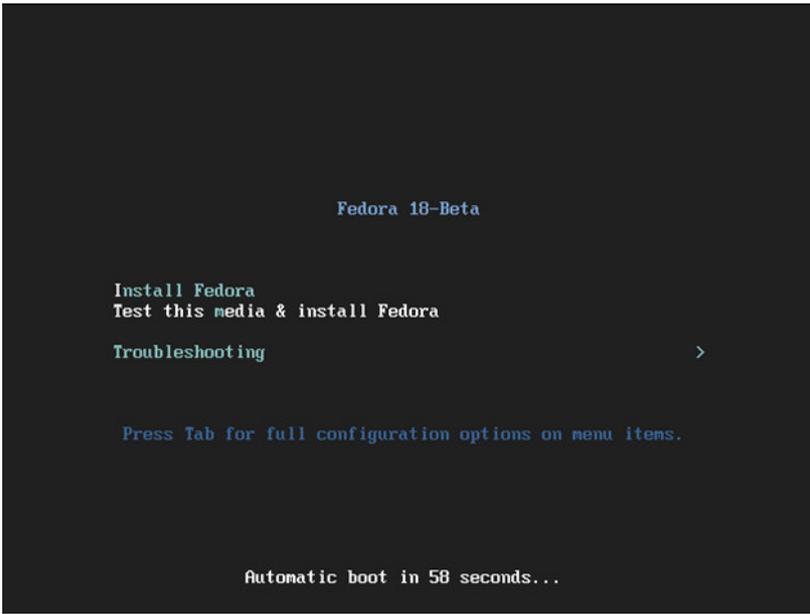


图 3.15 安装界面

(2) 经过一系列的自检之后，进入语言选择界面，这里建议使用 English (United States)，因为英文系统的查考资料较多，为以后驱动开发有一定的帮助。当然，如果你不习惯英文，选择简体中文也可以。语言选择界面如图 3.16 所示。

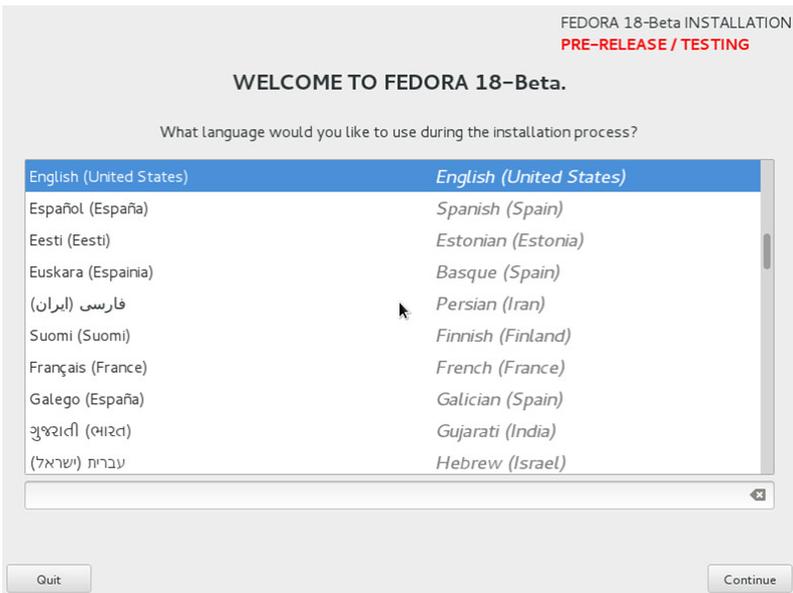


图 3.16 检查界面

(3) 当语言选择完之后，就会进入图形安装界面，这里的安装方法和 Windows 的安装方法类似。在安装过程中，用户可以选择键盘类型（一般为 U.S.English 式键盘）、安装程序、网络地址等。安装过程较为简单，用户可以根据提示进行选择和设置，这里就不详细讲解了。

3.1.3 设置共享目录

在网络连接畅通的情况下，虚拟机和 Windows 之间可以通过共享文件，完成两个系统的通信。设置共享文件，需要在 Windows 中设置共享文件夹，而且还需要在虚拟机上进行一些设置，这个过程如下。

(1) 在 Windows 系统中设置共享文件夹 share，右击文件夹，在弹出的快捷菜单中选择“共享此文件夹”单选按钮，如图 3.17 所示。

(2) 在虚拟机中设置网络连接（Network connection）为 Bridged 方式，这种方式可以使同一台机器上的两个操作系统之间进行通信，设置如图 3.18 所示。

(3) 在 Fedora 9 中，打开 Connect to Server 对话框，填写相应的服务器 IP 地址、共享文件夹、用户名和密码就能够访问 Windows 上的共享文件夹，如图 3.19 所示。

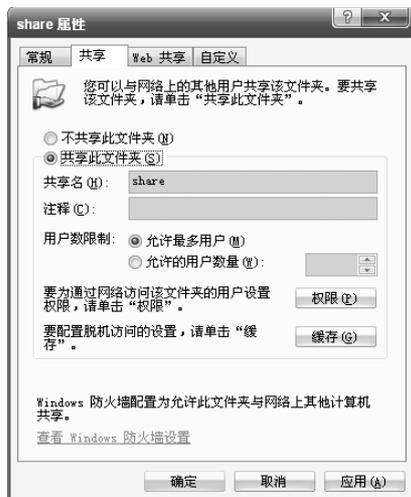


图 3.17 设置共享属性

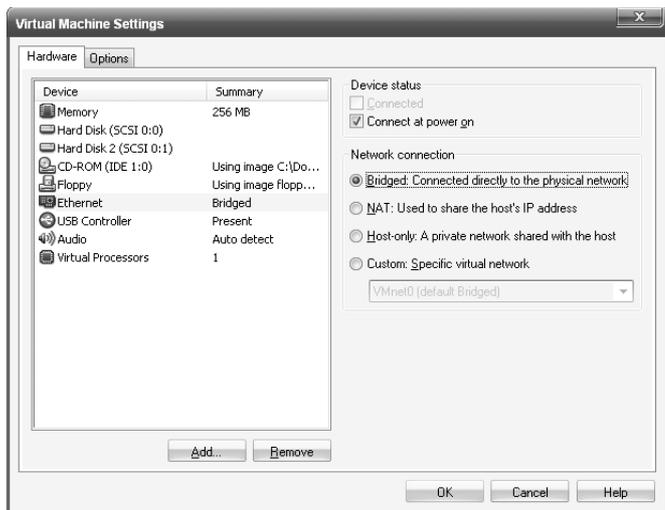


图 3.18 网络设置



图 3.19 共享连接

3.2 代码阅读工具 Source Insight

单独用一节来讲解代码阅读工具是否值得，答案是值得。因为 Linux 内核有 800 多万行代码，其中驱动程序占了 2/3 以上。阅读和理解这些代码，对编写设备驱动程序来说是非常有帮助的，所以本节将告诉大家怎样有效地使用代码阅读工具阅读代码。

3.2.1 Source Insight 简介

Source Insight 是一个非常好的代码阅读、编辑和分析的工具。Source Insight 支持目前大多数流行编程语言，如 C、C++、ASM、PAS、ASP 和 HTML 等。这个软件还支持关键字定义，对开发人员来说是非常有用的。Source Insight 不但能够编写程序，有代码自动提示的功能，而且还能够显示引用树、类图结构和调用关系等。

在分析 Linux 内核源代码时，使用这个软件可以很轻松地在代码之间跳转，并且捕获代码之间的关系。在程序员编写代码的时候，软件可以立刻分析源代码的信息，并显示给程序员。读者可以在 <http://www.sourceinsight.com> 上下载一个试用版本，这个版本可以使用 30 天。下面以分析 Linux 内核源代码为例，详细讲解 Source Insight 的使用。

3.2.2 阅读源代码

1. 建立 Source Insight 工程

Source Insight 默认情况下，只支持*.c 和*.h 文件，而 Linux 源代码中有大部分以“.S”结尾的汇编语言文件，所以需要设置一下 Source Insight 软件，使其支持“.S”文件。启动 Source Insight，选择 Options|Document Options 命令，打开 Document Options 窗口，如图 3.20 所示。选择 Document Type 的类型为 C Source File，并在 File filter 文本框中添加“.S”类型，使其支持汇编语言。

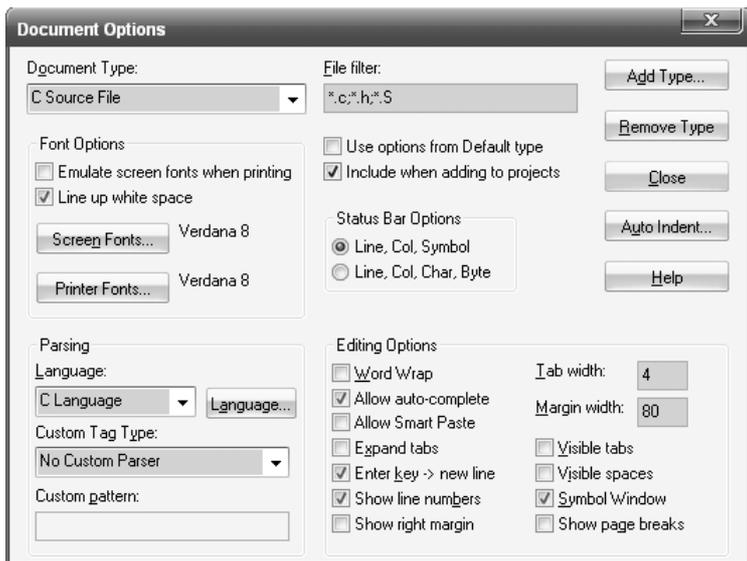


图 3.20 文件类型设置

需要建立一个新工程，将代码添加到工程中。首先选择菜单 Project|New Project 命令，建立一个新工程，如图 3.21 所示。

在随后弹出的对话框中，输入工程的名字和工程数据文件的存放位置。例如，在本例中工程的名称是 Linux2.6.34，数据文件存放在 F:\Source Insight\Projects\Linux2.6.34 目录下，如图 3.22 所示。

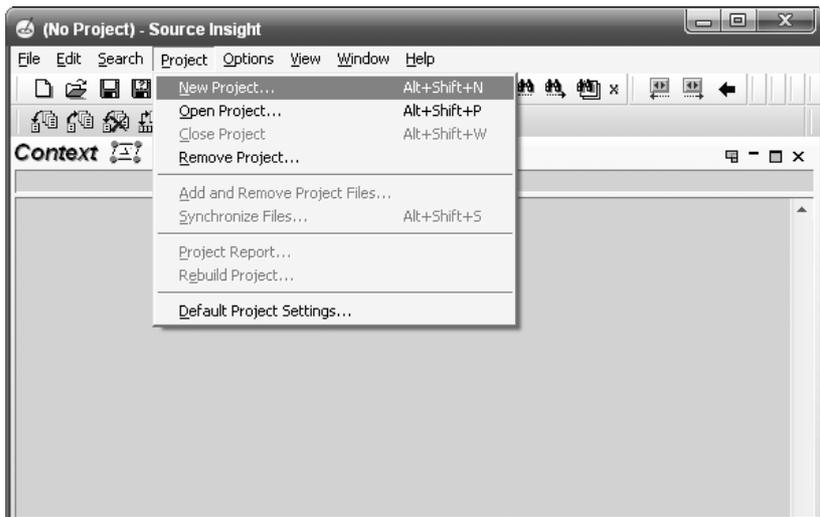


图 3.21 新建一个工程

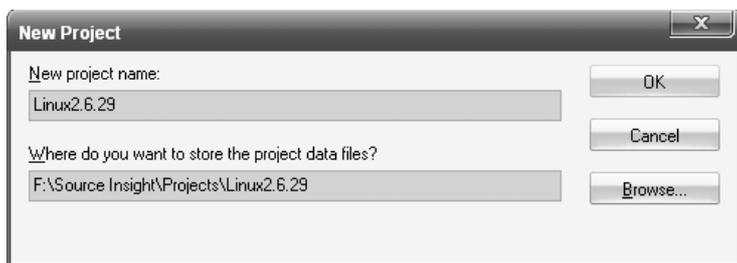


图 3.22 新建一个工程

单击 OK 按钮，进入指定需要分析的源代码位置对话框，如图 3.23 所示。在本例中，内核源代码的目录是 D:\linux2.6.34，指定这个目录。单击 OK 按钮进入下一个设置对话框。



图 3.23 指定待分析内核代码位置

这一步是表示添加哪些源文件到工程中，如图 3.24 所示。首先单击 Add All 按钮，在 Add to Project 对话框中，选择 Include top level sub-directories 和 Recursively and lower sub-directories 复选框。分别表示加入第一层子目录中的文件和递归的加入所有子目录中的文件。然后单击 OK 按钮将代码加入工程中，这样 Source Insight 工程就建立好了。

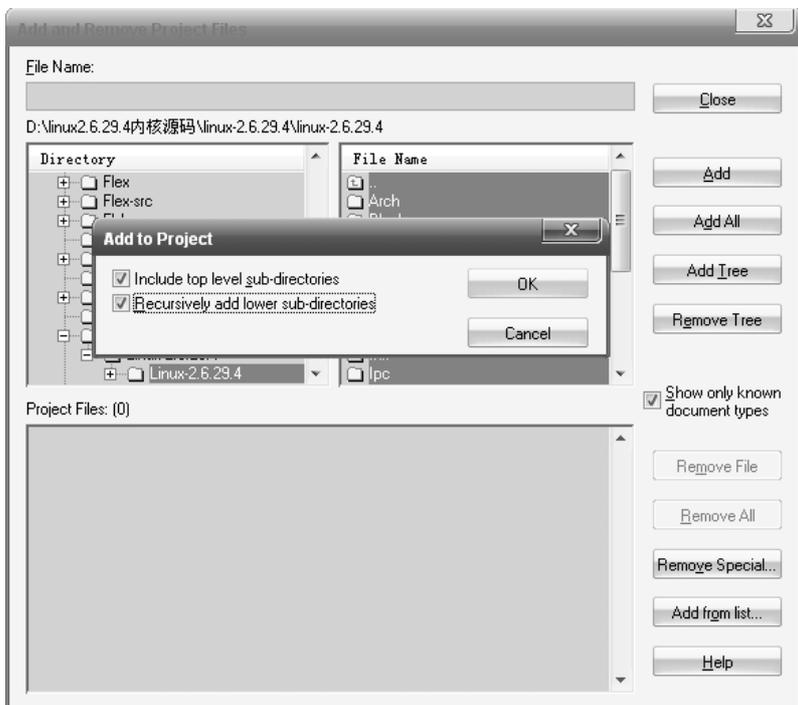


图 3.24 添加待分析文件到工程中

2. 更新数据库

Source Insight 的好处是可以给所有源文件中的各个变量、函数建立关系。这些关系被存储在了工程对应的数据库中。对于小型的工程，数据文件会自动建立。对于大型的工程，也就是像 Linux 内核源代码这样的工程，在使用时不能自动建立数据库。这个数据库较大，需要手动建立。

选择菜单 Project|Synchronize Files，打开 Synchronize Files 对话框，如图 3.25 所示。在其中选择 Force all files to be re-parsed 复选框，表示强制分析所有文件，为所有文件建立数据库。

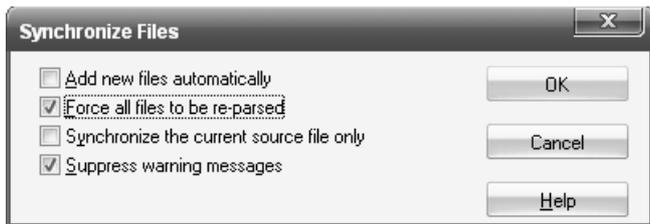


图 3.25 更新数据库

3. Source Insight 使用示例

Source Insight 的使用非常简单。如图 3.26 所示，在左下方的文件选择框中打开一个文件，例如 `irqflags.h` 文件。这个文件的内容显示在下面的主窗口中。可以在这个文件中找到一个 `__raw_local_irq_restore()` 函数。在主窗口中，按下 `Ctrl` 键，并单击 `__raw_local_irq_restore()` 函数就可以跳转到函数定义的位置。

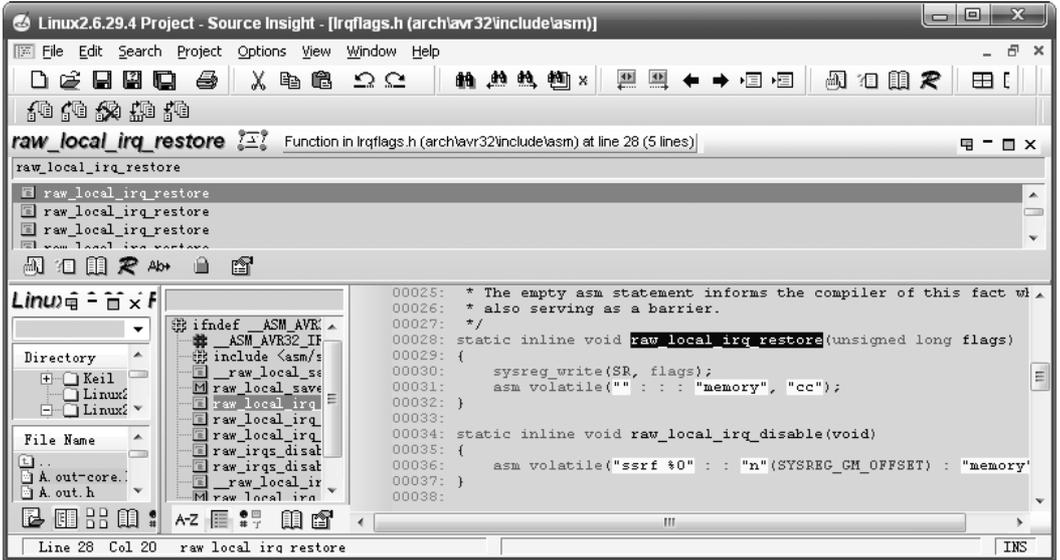


图 3.26 Source Insight 的界面

3.3 小 结

本章简要地介绍了驱动程序开发的一般环境，主要介绍了虚拟机和 Linux 操作系统的安装。另外，在驱动程序开发过程中，Windows 系统和 Linux 操作系统之间的数据传输也非常重要，所以本章也介绍了文件共享的方法。最后本章还介绍了一个分析和阅读源代码的工具，在实际的应用中非常有用。

第 4 章 构建嵌入式 Linux 操作系统

目前流行的嵌入式操作系统有 Linux、WinCE、VxWorks 等。Linux 作为一种免费的类 UNIX 操作系统，由于其功能强大，在嵌入式产品的应用中非常广泛。本章将对 Linux 操作系统做简单的介绍，并简述怎么构建一个可以运行的 Linux 操作系统。我相信，当读者自己构建出一个操作系统，将是一件非常愉悦的事情。

4.1 Linux 操作系统的介绍

Linux 操作系统是嵌入式系统的主流操作系统，本节对 Linux 操作系统进行简要的介绍，同时对 Linux 操作系统适用于嵌入式系统的原因进行简要分析。

4.1.1 Linux 操作系统

Linux 操作系统是一个类 UNIX 操作系统。Linux 操作系统内核的名字也是 Linux。Linux 这个词本身只表示 Linux 内核，但在实际中人们已经习惯了用 Linux 形容整个基于 Linux 内核的操作系统。Linux 的最初版本由 Linus Torvalds 开发，此后得到互联网上很多计算机高手的支持，Linux 的发展之迅速，实在让人惊叹不已，目前的版本已经到了 3.8，已经是一个非常成熟稳定的操作系统。下面从不同方面对 Linux 操作系统进行简要的介绍。

1. Linux 的诞生

Linux 诞生于一位名叫 Linus Torvalds 的计算机业余爱好者，当时他是芬兰赫尔辛基大学的学生。他开发 Linux 的最初目的是想设计一个代替 Minix (Minix 是由一位名叫 Andrew Tannebaum 的计算机教授编写的一个操作系统教学程序) 的操作系统。Minix 这个操作系统可用于 386、486 或奔腾处理器的个人计算机上，并且具有 UNIX 操作系统的大部分功能。由于 Andrew Tannebaum 教授并不允许开发人员对 Minix 进行扩展，所以 Linus Torvalds 决定开发一个新的类似于 Minix 的操作系统，但相比 Minix 有更多的功能。Andrew Tannebaum 教授并不允许开发人员对 Minix 进行扩展的原因是，他想维持 Minix 的简单性，使其更利于教学，有很多学生从中受益，当然反过来，这个限制也局限了 Minix 的发展。

2. Linux 与 GNU 计划

Linux 的发展与 GNU 计划密切相关。1983 年, Richard Stallman 创立了 GNU 计划(GNU Project)。这个计划有一个目标，是为了发展一个完全免费自由的类 UNIX 的操作系统。自 1990 年发起这个计划以来，GNU 开始大量地收集和开发类 UNIX 系统所必备的元件，例如函数库 (libraries)、编译器 (compilers)、调试工具 (debuggers)、文字编辑器 (text editors)、

网页服务器 (web server), 以及一个 UNIX 的用户接口 (Unix shell), 但是一个好的内核核心一直没有出现。

1990 年, GNU 计划开始在 Mach microkernel 的架构之上开发内核核心, 也就是所谓的 GNU Hurd 计划, 但是这个基于 Mach 的设计异常复杂, 发展进度相对缓慢, 并没有取得太大的成效。恰好此时, 大约是 1991 年 4 月, Linus Torvalds 开发的 Linux 0.01 版被他发布到互联网上, 引起了很多程序员的关注。

Linus Torvalds 宣布这是一个免费的系统, 主要在 x86 电脑上使用。Linus Torvalds 希望大家一起来完善它, 并将源代码放到了芬兰的 FTP 站点上任人免费下载。本来他想把这个系统称为 freax, 意思是自由 (free) 和奇异 (freak) 的结合字, 并且附上了 X 这个常用的字母, 以配合所谓的类 UNIX (Unix-like) 的系统。可是 FTP 的工作人员认为这是 Linus 的新操作系统, 觉得原来的命名 Freax 的名称不好听, 就用 Linux 这个子目录来存放, 于是大家就将它称为 Linux。这时的 Linux 只有内核程序, 仅有 10000 行代码, 仍必须执行于 Minix 操作系统之上, 并且必须使用硬盘开机, 还不能称做是完整的操作系统; 随后在 10 月份 Linux 的第二个版本 (0.02 版) 发布, 许多专业程序员自愿地开发它的应用程序, 并借助 Internet 拿出来让大家一起修改。在很短的一段时间内, Linux 的应用程序越来越多, 由此 Linux 本身也逐渐发展壮大起来。到目前为止最新的内核主版本已经是 3.8 了。

4.1.2 Linux 操作系统的优点

Linux 操作系统有很多优点, 具有十分丰富的应用功能。这些功能特别适用于嵌入式系统, 这些优点如下。

1. 价格低廉

Linux 操作系统使用了大量的 GNU 软件, 包括 shell 程序、工具集、程序库、编译器等。这些程序都可以免费或者以极低的价格得到, 所以 Linux 操作系统是一个价格低廉的操作系统。基于这个原因, Linux 常常被应用于嵌入式系统中, 例如机顶盒、移动电话甚至机器人中。在移动电话上, 基于 Linux 的 Android 已经成为与 Windows8 系统并列的三大智能手机操作系统之一; 而在移动装置上, 则成为 Windows CE 与 Palm OS 外另一个好的选择。此外, 还有不少硬件式的网络防火墙及路由器, 其内部都是使用 Linux 操作系统, 其执行效率和安全性非常高。

2. 高效性和灵活性

Linux 以它的高效性和灵活性著称。Linux 操作系统是一个非常高效的系统, 广泛应用于对效率要求较好的服务器上。另外, Linux 操作系统的灵活性也是其他操作系统无法比拟的。Linux 操作系统可以根据用户需要自己配置内核, 增加或者减少相应的功能。通过这种方式, Linux 操作系统几乎支持目前所有的常用硬件, 就算有不支持的硬件, 驱动开发人员也可以在很短的时间内写出相应的驱动程序来。

3. 广泛性

Linux 操作系统可以应用于目前大多数处理器架构上, 其应用非常广泛。据统计, 目

前世上运行最快的 500 台超级计算机上，有 74% 的计算机使用的都是 Linux 操作系统。对于嵌入式系统，处理器的选择非常广泛，幸运的是，Linux 几乎支持所有的主流处理器，最典型的就是 ARM 处理器。嵌入式系统开发人员可以直接移植 Linux 操作系统，并选择一些可靠的自由软件就能够组装一个有用的嵌入式系统，极大地减少了开发时间。

4. 强大的功能

每一天，全球有很多开发人员都在对 Linux 操作系统进行开发，所以每一天都有新的功能被添加到 Linux 中。到目前为止，Linux 已经发展成了一个遵循 POSIX 标准的纯 32 位操作系统，64 位版本也已经发布。Linux 可以兼容大部分的 UNIX 系统，很多 UNIX 的程序不需要改动，或者很少的改变就可以运行于 Linux 环境中；内置 TCP/IP 协议，可以直接连入 Internet，作为服务器或者终端使用；内置 Java 解释器，可直接运行 Java 源代码；具备程序语言开发、文字编辑和排版、数据库处理等能力；提供 X Window 的图形界面；主要用于 x86 系列的个人电脑，也有其他不同硬件平台的版本，支持现在流行的所有硬件设备。

就性能上来说，它并不弱于 Windows 甚至 UNIX，而且靠仿真程序还可以运行 Windows 应用程序。它有成千上万的各类应用软件，并不输于 Windows 的应用软件数量，其中也有商业公司开发的赢利性的软件。

4.2 Linux 内核子系统

编写设备驱动程序，涉及 Linux 内核的许多子系统，了解这些子系统对于了解 Linux 操作系统和编写设备驱动程序都非常有用。这些主要的子系统包括进程管理、内存管理、文件管理、设备管理和网络管理。现对这些主要的子系统分别介绍如下。

4.2.1 进程管理

进程是操作系统中一个很重要的概念。进程是操作系统分配资源的基本单位，也是 CPU 调度的基本单位。可以给进程这样一个定义：进程是程序运行的一个实例，是操作系统分配资源和调度的一个基本单位。Linux 将进程分为就绪状态、执行状态和阻塞 3 个状态。Linux 内核负责对这 3 种状态进行管理。下面对这 3 种状态的基本概念介绍如下。

- ❑ 就绪状态：在这种状态中，进程具有处理器外的其他资源，进程不运行。当处理器空闲时，进程就被调度来运行。
- ❑ 执行状态：进程处于就绪状态后，获得处理器资源，就能进入执行状态，此时程序正在运行。
- ❑ 阻塞状态：进程因为等待某种事件的发生而暂时不能运行。这些事件如设备中断，其他进程的信号。这 3 种状态的状态转换如图 4.1 所示。

如图 4.1 所示，当系统分配资源并创建一个进程后，进程就进入就绪状态。当调度程序分配了处理器资源后，进程便进入执行状态。相应地，当处理器资源用完后，进程又进入就绪状态。在执行状态中，因发生某些事件而使进程不能运行时，则进程进入阻塞状态。在阻塞状态下，当外部事件得到满足后，进程就进入就绪状态。进行调度看上去似乎很复

杂，但本质上是进程争夺 CPU 的过程。这就像在售票窗口买火车票一样，买票之前，你必须排队，这就是就绪状态。售票员给你卖票，这就是执行状态。如果在卖票的过程中，你的身份证还没有拿出来，钱也没有拿出来，不好意思，请先准备好再来买票，这个时候，售票员就把你踢到了阻塞状态。当你把钱、身份证都准备好后，那么你就继续排队买票吧，这样你又进入了就绪状态。

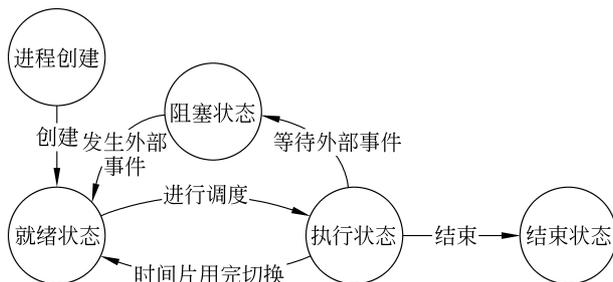


图 4.1 进程的状态切换

4.2.2 内存管理

内存是计算机的主要资源之一，可以将内存理解为一个线性的存储结构。用来管理内存的策略是决定系统性能的主要因素。内核在有限的资源上为每一个进程创建一个虚拟地址空间，并对虚拟地址空间进行管理。为了方便内存的管理，内核提供了一些重要的函数。这些函数包括 `kmalloc()` 和 `kfree()` 等。另外设备驱动程序需要使用内存分配，不同的分配方式对驱动程序的影响不同，所以需要内存分配有比较清晰的了解。

4.2.3 文件系统

在 Linux 操作系统中，文件系统是用来组织、管理、存放文件的一套管理机制。Linux 文件系统的一大优点是，它几乎可以支持所有的文件格式。任何一种新的文件格式，都可以容易地写出相应的支持代码，并无缝地添加到内核中。虽然不同文件格式的文件以不同的存储方式存放在磁盘设备中，但是在用户看来，文件总以树形结构显示给用户。这种树形结构如图 4.2 所示。

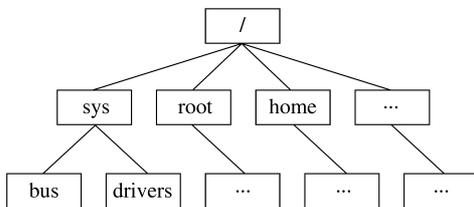


图 4.2 文件的树形结构

另一个方面，在 Linux 中，几乎每一个对象都可以当作文件来看待，最常见的就是设备文件。设备文件将设备当作文件来看待，这样就可以像操作文件一样操作设备，也就是可以使用 `read()` 和 `write()` 等函数来读取数据

4.2.4 设备管理

无论是桌面系统还是嵌入式系统，都存在各种类型的设备。操作系统的一个重要功能就是对这些设备进行统一的管理。由于设备的种类繁多，不同设备的操作方法都不一样，使管理设备成为操作系统中非常复杂的部分。Linux 系统通过某种方式较好地解决了这个问题，使设备的管理得到了统一。

设备管理的一个主要任务是完成数据从设备到内存的传输。一个完整的数据传输过程是数据首先从设备传入内存，然后 CPU 对其进行处理，处理完后将数据传入内存或设备中。

4.2.5 网络功能

网络功能也由操作系统来完成。大部分的网络操作与用户进程都是分离的，数据包的接收和发送操作都是由相应的驱动程序来完成的，而与用户进程无关。进程处理数据之前，驱动程序必须先收集、标识和发送或重组数据。当数据准备好后，系统负责用户进程和网络接口之间的数据传送。另外内核也负责实现网络通信协议。

4.3 Linux 源代码结构分析

了解 Linux 源代码结构对理解 Linux 如何实现各项功能是非常重要的。对驱动程序的编写也非常重要，这样，驱动开发人员知道应该在何处找到相关的驱动程序，一方面可以对其进行修改移植，另一个方面可以模仿以往的驱动程序，写出新的驱动程序。Linux 源代码以目录的方式组织，每一个目录中有相关的内核代码。下面对各个主要的目录进行介绍。

4.3.1 arch 目录

随着 Linux 操作系统的广泛应用，特别是 Linux 在嵌入式领域的发展，越来越多的人开始投身到 Linux 驱动开发中。面对日益庞大的 Linux 内核源代码，驱动开发者在完成自己的内核代码后，都将面临着同样的问题，即如何将源代码融入到 Linux 内核中，增加相应的 Linux 配置选项，并最终被编译进 Linux 内核。这就需要对 Linux 源代码结构进行详细的介绍，首先介绍 arch 目录。

arch 目录中包含与体系结构相关的代码，每一种平台都有一种相应的目录，常见目录如表 4.1 所示。

表 4.1 arch 目录

一级目录	二级目录	说 明
Arch	Alpha	康柏的 Alpha 体系结构计算机
	Arm	基于 Arm 处理的体系结构，此目录中包含支持 Arm 处理器的代码
	Arv32	Arv 体系结构的计算机
	Cris	Cris 体系结构的计算机
	frv	frv 体系结构的计算机
	X86	IBM 的 PC 体系结构计算机

4.3.2 drivers 目录

drivers 目录中包含了 Linux 内核支持的大部分驱动程序。每种驱动程序都占用一个子目录。目录中包含了驱动的大部分代码，这些目录和目录的功能如表 4.2 所示。

表 4.2 drivers 目录

一级目录	二级目录	说 明
drivers	ftape	磁带驱动
	hfmodem	无线电设备驱动
	joystick	游戏杆驱动
	paride	从并口访问 IDE 设备的支持
	acorn	Acorn 设备驱动
	ap1000	富士的 AP1000
	cdrom	光驱驱动
	char	字符设备驱动程序
	fc4	光纤设备
	misc	杂项设备驱动
	net	网卡驱动
	Pci	PCI 总线驱动
	scsi	SCSI 设备驱动
	sound	音频设备驱动
	usb	usb 串行总线驱动
video	视频卡设备驱动	
block	块设备驱动	

4.3.3 fs 目录

fs 目录中包含了 Linux 所支持的所有文件系统相关的代码。每一个子目录中包含一种文件系统，例如 msdos 和 ext3。Linux 几乎支持目前所有的文件系统，如果发现一种没有支持的新文件系统，那么可以很方便地在 fs 目录中添加一个新的文件系统目录，并实现一种文件系统。fs 目录的详细内容如表 4.3 所示。

表 4.3 fs 目录

一级目录	二级目录	说 明
fs	Adfs	Acorn 磁盘填充文件系统
	Affs	Amiga 快速文件系统 (FFS)
	Autofs	支持自动装载文件系统的代码
	Coda	Coda 网络文件系统
	Devpts	/dev/pts 虚拟文件系统
	efs	SGIIRIX 公司的 EFS 文件系统
	ext2	Linux 支持的 Ext2 文件系统

续表

一级目录	二级目录	说 明
fs	fat32	Windows 支持的 Fat 文件系统
	Hfs	苹果的 Macintosh 文件系统
	hpfs	IBM 的 OS/2 文件系统
	isofs	ISO9660 文件系统（光盘文件系统）
	minix	MINIX 文件系统，MINIX 系统的文件系统
	msods	微软的 MS-DOS 文件系统
	ncpfs	Novell 的 Netware 核心协议
	nfs	一种网络文件系统
	ntfs	微软的 WindowsNT 文件
	proc	/proc 文件系统
	romfs	只读文件系统，只存在于内存中
	smbfs	微软的 SMB 服务器文件系统
	ufs	Linux 的一种文件系统
	umsdos	UMSDOS 文件系统
vfat	微软的 VFAT 文件系统	

4.3.4 其他目录

除了上面介绍的目录外，内核中还有其他一些重要的目录和文件。每一个目录和文件都有自己特殊的功能，下面对这些目录和文件进行简要的介绍。

表 4.4 其他目录

目录或者文件	说 明
include	该目录包含编译内核所需的大部分头文件。其子目录/include/linux 中，包含与平台无关的头文件，与平台有关的头文件放在各自的单独目录中
init	内核的初始化代码，包含系统启动的 main() 函数
ipc	该目录包含进程间通信的代码
kernel	内核最核心的代码，包括进程调度、内存管理等
lib	该目录包含库模块代码
mm	该目录包含独立于 CPU 体系结构的内存管理代码。不同平台的代码在该目录下有相应的目录
net	包含各种网络协议
scripts	包含一些脚本文件，内核配置相关的文件
security	一个 SELinux 模块
sound	常用的音频设备驱动程序
usr	一个 cpio
block	块设备驱动程序
crypto	常用的加密和压缩算法
Documentation	内核部分功能的解释文档

目录或者文件	说 明
COPYING	GPL 版权声明文件
CREDITS	内核开发者列表, 包含对 Linux 做出很大贡献的人的信息
Kbulid	用来编译内核的脚本
MAINTAINERS	维护人员列表
Makefile	第一个 Makefile 文件, 用来组织内核的各个模块, 记录了各个模块相互之间的联系。编译器根据这个文件来编译内核
Readme	内核及编译方法的介绍
REPORTING-BUGS	关于 bug 的一些内容

Linux 内核源代码的学习是一个长期的过程, 在以后的深入学习中, 相信读者能够对内核源代码有更深入的理解。

4.4 内核配置选项

自己构建嵌入式 Linux 操作系统, 首先需要对外核源代码进行相应的配置。这些配置决定了嵌入式 Linux 操作系统所支持的功能, 为了理解编译程序是怎样通过配置文件配置系统的, 下面对配置编译过程进行详细的讲解。

4.4.1 配置编译过程

面对日益庞大的 Linux 内核源代码, 要手动地编译内核是十分困难的。幸好 Linux 提供了一套优秀的机制, 简化了内核源代码的编译。这套机制由以下几方面组成。

- ❑ **Makefile 文件:** 它的作用是根据配置的情况, 构造出需要编译的源文件列表, 然后分别编译, 并把目标代码链接到一起, 最终形成 Linux 内核二进制文件。由于 Linux 内核源代码是按照树形结构组织的, 所以 Makefile 也被分布在目录树中。
- ❑ **Kconfig 文件:** 它的作用是为用户提供一个层次化的配置选项集。make menuconfig 命令通过分布在各个子目录中的 Kconfig 文件构建配置用户界面。
- ❑ **配置文件 (.config):** 当用户配置完后, 将配置信息保存在.config 文件中。
- ❑ **配置工具:** 包括配置命令解释器 (对配置脚本中使用的配置命令进行解释) 和配置用户界面 (提供基于字符界面、基于 Ncurses 图形界面以及基于 Xwindows 图形界面的用户配置界面, 各自对应于 Make config、Make menuconfig 和 make xconfig)。

这套机制在目录中的位置如图 4.3 所示。

从图 4.3 中可知, 主目录中包含很多子目录, 同时包含 Kbulid 和 Makefile 文件。各子目录中也包含其他子目录和 Kbulid 和 Makefile 文件, 只是图中不好画出。当执行 menuconfig 命令时, 配置程序会依次从目录由浅入深查找每一个 Kbulid 文件, 依照这个文件中的数据生成一个配置菜单。从这个意义上来说, Kbulid 像是一个分布在各个目录中的配置数据库, 通过这个数据库可以生成配置菜单。在配置菜单中根据需要配置完成后会在主目录下生成一个.config 文件, 此文件中保存了配置信息。

然后执行 make 命令时, 会依赖生成的.config 文件, 以确定哪些功能将编译入内核中, 哪些功能不编译入内核中。然后递归地进入每一个目录, 寻找 Makefile 文件, 编译相应的

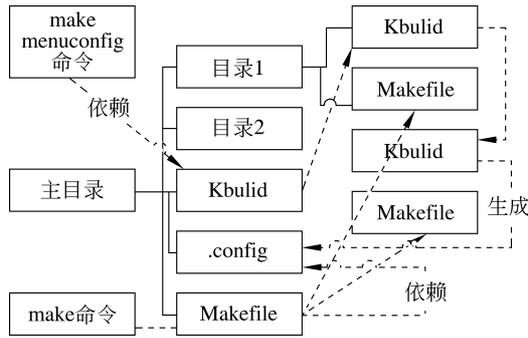


图 4.3 配置文件的组织关系和编译过程

代码。这个过程在图 4.3 中有很清晰的显示。

4.4.2 常规配置

常规配置包含关于内核的大量配置，这些配置包含代码成熟度、版本信息和模块配置等。下面分别介绍。

1. 常规配置选项

常规配置包含了一些通用配置，主要与进程相关，例如进程的通信和进程的统计等。这些配置的详细信息如表 4.5 所示。

表 4.5 常规配置选项

第一级配置选项	第二级配置选项	说 明
Code maturity level options	Prompt for development and/or incomplete code/drivers	显示尚在中或尚未完成的代码与驱动，除非你是测试人员或者开发者，否则请勿选择
General setup	Local version - append to kernel release	在内核版本字符串后面加上一个自定义的版本字符串（小于 64 字符），可以用“ <code>uname -a</code> ”命令看到这个字符串
	Automatically append version information to the version string	是否在版本字符串后面添加版本信息，编译时需要有 perl 及 git 库支持
	Support for paging of anonymous memory (swap)	允许虚拟内存使用交换文件或者交换分区
	System V IPC	允许进程间通信（IPC），大多数程序需要这个功能，所以为必选
	IPC Namespaces	IPC 命名空间支持，不重要
	POSIX Message Queues	支持 POSIX 消息队列
	BSD Process Accounting	支持将进程的统计信息写入文件，包括进程的创建时间/创建者/内存占用等信息
	BSD Process Accounting version 3 file format	使用新的第三版文件格式，该文件格式请查阅相关资料
	Export task/process statistics through netlink	通过 netlink 接口向用户空间导出任务/进程的统计信息，与 BSD Process Accounting 的不同之处在于这些统计信息在整个任务/进程生存期都是可用的
UTS Namespaces	UTS 名字空间支持，一般不选	